PhoneNet使用说明

|  |  |  |  |
| --- | --- | --- | --- |
| **版本号** | **修订日期** | **修订内容** | **说明** |
| V1.0 | 2019.07.19 | 新增 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[PhoneNet使用说明 1](#_Toc30695)

[1 说明 3](#_Toc32085)

[2 接口函数 3](#_Toc7403)

[3 事件回调接口 6](#_Toc1210)

[4 DEMO说明 7](#_Toc14091)

# 说明

* 本文档是PhoneNet库的使用教程，基于C#开发，适合.Net开发用户;
* 新建项目时，需要引用PhoneNet.dll库，我们的DEMO项目已经引用了，无需再引用；

# 接口函数

PhoneNet库的PhoneSdk类包含了所有的VOIP相关的接口；

/// <summary>

/// 初始化SDK

/// </summary>

/// <param name="context">主线程的Context</param>

/// <returns></returns>

public int InitSdk(System.Threading.SynchronizationContext context)

初始化SDK函数必须最先调用，且只能调用一次；

如：

private PhoneSdk m\_Phone = PhoneSdk.Instance; // PhoneSdk是单列，通过此属性返回；

m\_Phone.InitSdk(new WindowsFormsSynchronizationContext());

/// <summary>

/// 获取SDK版本号

/// </summary>

/// <returns></returns>

public int GetSdkVersion()

/// <summary>

/// 注册到SIP服务器

/// </summary>

/// <param name="serverAddr">服务器地址</param>

/// <param name="port">服务器端口</param>

/// <param name="userID">用户名</param>

/// <param name="userPWD">密码</param>

/// <param name="accountID">账户ID，若只有一个账户写0</param>

/// <param name="domain">服务器域名</param>

/// <param name="authWithDomain">是否带域名认证，默认为false。参考服务器的要求</param>

/// <param name="idUri">参考服务器的注册要求，默认为空</param>

/// <param name="useIMS3GPP">是否采用IMS的3gpp认证，参考服务器的注册要求</param>

/// <param name="extraHeads">扩展消息格式标准为"X-HEAD\_NAME1:head1 value;X-HEAD\_NAME2:head2 value",头部消息间用";"分隔:

/// 如:"X-SparkleComm-Call-DEVICE-ID: EF32D-98AF-9E01-882E;X-SparkleComm-Call-MSG2:this is other message"</param>

public void RegisterSipServer(string serverAddr, int port, string userID, string userPWD, int accountID,

string domain, bool authWithDomain, string idUri, bool useIMS3GPP, string extraHeads)

如：m\_Phone.RegisterSipServer("192.168.0.247", 5060, "1019", "123456", 0, "192.168.0.247", false, "", false, "");

/// <summary>

/// 注销SIP账户

/// </summary>

/// <param name="accountID">对应RegisterSipServer里面的accountID</param>

public void UnRegister(int accountID = 0)

若更改了SIP相关信息时，建议先调用UnRegister，再调用RegisterSipServer重新注册新的SIP账户。

/// <summary>

/// 拨打电话

/// </summary>

/// <param name="calleeAdd">对方号码</param>

/// <param name="extraData">扩展消息格式标准为"X-HEAD\_NAME1:head1 value;X-HEAD\_NAME2:head2 value",头部消息间用";"分隔:如:"X-SparkleComm-Call-DEVICE-ID: EF32D-98AF-9E01-882E;X-SparkleComm-Call-MSG2:this is other message"</param>

/// <param name="accountID">呼叫帐号编号(用于多账户注册)</param>

/// <returns>通话ID，成功返回>=0，失败为-1</returns>

public int MakeVoiceCall(string calleeAdd, string extraData = "", int accountID = 0)

/// <summary>

/// 释放电话：挂断电话

/// </summary>

/// <param name="accountID">当前注册的账户ID</param>

/// <param name="callID">当前通话的ID</param>

public void ReleaseCall(int accountID = 0, int callID = -1)

/// <summary>

/// 开始录音

/// </summary>

/// <param name="fileName">录音文件</param>

/// <param name="accountID"></param>

/// <param name="callID">通话ID</param>

/// <returns>录音ID：成功大于0</returns>

public int StartRecord(string fileName, int accountID, int callID)

/// <summary>

/// 停止录音

/// </summary>

/// <param name="recordId">录音ID</param>

/// <param name="accountID">账户ID</param>

/// <param name="callID">通话ID</param>

/// <returns>是否成功</returns>

public bool StopRecord(int recordId, int accountID, int callID)

/// <summary>

/// 设置呼叫保持

/// </summary>

/// <param name="accountID">账户ID</param>

/// <param name="callID">通话ID</param>

public void CallHold(int accountID, int callID)

/// <summary>

/// 取消呼叫保持

/// </summary>

/// <param name="accountID"></param>

/// <param name="callID"></param>

public void ReleaseHold(int accountID, int callID)

/// <summary>

/// 接听电话

/// </summary>

/// <param name="accountID"></param>

/// <param name="callID"></param>

public void AnswerCall(int accountID, int callID)

/// <summary>

/// 发送DTMF消息

/// </summary>

/// <param name="text"></param>

public void SendDtmf(string text)

/// <summary>

/// 设置静音

/// </summary>

/// <param name="enable">true=静音；false=取消静音</param>

public void SetMicMute(bool enable)

/// <summary>

/// 呼叫转移

/// <param name="targetAddr">目标地址</param>

/// <param name="accountID"></param>

/// <param name="callID"></param>

public void CallTransfer(string targetAddr, int accountID, int callID)

/// <summary>

/// 获取通话信息

/// </summary>

/// <param name="calls"></param>

/// <param name="count"></param>

public void GetAllCallInfo(out CallInfo[] calls, ref int count)

/// <summary>

/// 获取当前通话数量

/// </summary>

/// <returns></returns>

public int GetCallCount()

# 事件回调接口

PhoneSdk类包含了如下回调事件，根据事件进行相关业务操作，可以参考DEMO实现，所有的事件回调都是在主线程里运行的。

/// <summary>

/// SIP服务注册成功事件

/// </summary>

public event SipConnectedDelegate OnSipConnectedEvent;

/// <summary>

/// SIP服务器断开连接事件

/// </summary>

public event SipDisconnectedDelegate OnSipDisconnectedEvent;

/// <summary>

/// 信号强度事件

/// </summary>

public event SignalReportDelegate OnSignalReportEvent;

/// <summary>

/// 通话连接后，准备进行媒体协商的事件

/// </summary>

public event CallConnectEarlyDelegate OnCallConnectEarlyEvent;

/// <summary>

/// 通话连接后的媒体协商已经成功

/// </summary>

public event CallConnectMediaDelegate OnCallConnectMediaEvent;

/// <summary>

/// 通话已经连接成功

/// </summary>

public event CallConnectedDelegate OnCallConnectedEvent;

/// <summary>

/// 通话已经释放

/// </summary>

public event CallReleaseDelegate OnCallReleaseEvent;

/// <summary>

/// 来电事件

/// </summary>

public event CallIncomingDelegate OnCallIncomingEvent;

# DEMO说明

SoftPhoneCSharp目录是C#版本的DEMO工程；

frmMain.cs：主界面类，主要实现注册功能、来电消息；

frmIncoming.cs：来电界面；

frmDialer.cs：拨号界面；

frmInCall.cs：通话界面；